Frobenius correlation based u-shapelets discovery for time series clustering

Vanel Steve Siyou Fotso, Engelbert Mephu Nguifo, Philippe Vaslin

PII:	S0031-3203(20)30105-9
DOI:	https://doi.org/10.1016/j.patcog.2020.107301
Reference:	PR 107301
To appear in:	*Pattern Recognition*
Received date:	3 October 2018
Revised date:	14 December 2019
Accepted date:	23 February 2020

Please cite this article as: Vanel Steve Siyou Fotso, Engelbert Mephu Nguifo, Philippe Vaslin, Frobenius correlation based u-shapelets discovery for time series clustering, *Pattern Recognition* (2020), doi: https://doi.org/10.1016/j.patcog.2020.107301

- We review state of the art on similarity functions for uncertain time Series and evaluate them for the comparison of small, uncertain time series.

- We introduce the Frobenius cOrrelation for uncertain Time series ushapelet discovery (FOTS), a new dissimilarity score based on local correlation, which has interesting properties useful for comparison of small, uncertain time series and that makes no assumption on the probability distribution of uncertainty in data.

- We evaluate FOTS on 63 datasets on clustering task.

- We put the source code at the disposal of the scientific community to allow extension of our work.

Frobenius correlation based u-shapelets discovery for time series clustering

Vanel Steve Siyou Fotso[a,*], Engelbert Mephu Nguifo[a], Philippe Vaslin[a]

[a] *University Clermont Auvergne, CNRS, LIMOS, F-63000 Clermont-Ferrand, France*

Abstract

An u-shapelet is a sub-sequence of a time series used for the clustering of time series datasets. The purpose of this paper is to discover u-shapelets on uncertain time series. To achieve this goal, we propose a dissimilarity score called FOTS whose computation is based on the eigenvector decomposition and the comparison of the autocorrelation matrices of the time series. This score is robust to the presence of uncertainty; it is not very sensitive to transient changes; it allows capturing complex relationships between time series such as oscillations and trends, and it is also well adapted to the comparison of short time series. The FOTS score is used with the Scalable Unsupervised Shapelet Discovery algorithm for the clustering of 63 datasets, and it has shown a substantial improvement in the quality of the clustering with respect to the Rand Index. This work defines a novel framework for the clustering of uncertain time series.

Keywords: Clustering, UShapelet, Correlation, Time series

1. Introduction

All measurements performed by a mechanical system contain uncertainty. Indeed, the uncertainty principle is partly a statement about the limitations of mechanical systems ability to perform measurements on a system without disturbing it [1]. Thus, time series from measurement instruments *are uncertain.* These time series produced by sensors constitute a vast proportion of the time

[*] Corresponding author
Email address: siyou@isima.fr (Vanel Steve Siyou Fotso)

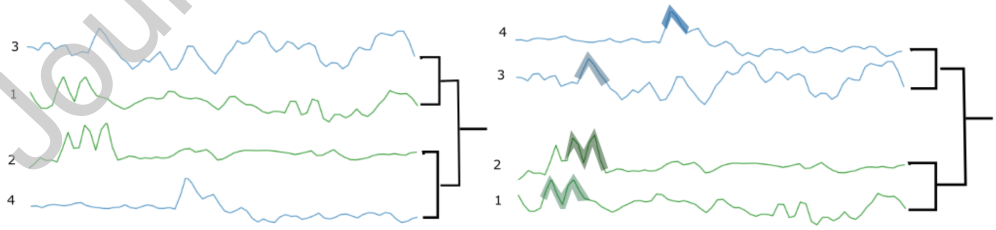
series used in science, whether in medicine with ECGs, in physics with measurements recorded by telescopes, in computing with the Internet of Things and so on. Ignoring the uncertainty of the data during their analysis can lead to inaccurate conclusions [2], hence the need to implement uncertain data management techniques.

Several recent studies have focused on the processing of uncertainty in data mining. Rizvandi et al.[3] studied CPU utilization time patterns of several MapReduce applications using Dynamic Time Warping and Euclidian distance for comparing times series, and they investigated the minimum distance/maximum similarity of these applications. Their results showed the effectiveness of their approach on a private cloud with up to 25 virtual nodes. Considering that time series data often contain uncertainty and that DUST is one of the latest methods that can deal with arbitrary probability distributions, but that its computational cost is high particularly when the dataset is large, Hwang et al. [4] demonstrated that the performance of DUST was much faster using GPU than the CPU-based implementation. Rehfeld and Kurths [5] investigated similarity estimators that could be suitable for the quantitative investigation of dependencies in irregular and age-uncertain time series like paleoclimate time series. They concluded that age uncertainty contributes up to half of the uncertainty in the similarity estimation process and that their new event synchronization function (ESF) could be suitable to study extreme event dynamics in paleoclimate records. Orang and Shiri [6] presented an overview of deterministic and probabilistic similarity measures and evaluate them experimentally on uncertain time series. Their results provided useful insights and guidelines for researchers and practitioners in similarity search and analysis of uncertain time series data. Orang and Shiri [7] formalized the notion of normalization and correlation for UTS in two general settings based on the available information at each timestamp (i.e. PDF-based UTS and multiset-based UTS) and, for each case, they developed techniques to determine the underlying probability density function. Their results demonstrated the effectiveness of the proposed techniques and the second one particularly showed a significant improvement in space utilization and computation time. The same authors [8] studied the impact of preprocessing techniques on performance and effectiveness of the similarity measures for uncertain time series. They showed that the performance of uncertain similarity measures can be improved through preprocessing techniques, which outperformed traditional similarity measures.

This literature analysis reveals that two main approaches allow to take uncertainty into account in data mining tasks: either during the comparison phase by using appropriate distance functions [3, 4, 5, 6, 9, 7], or its impact is reduced by transformations performed on the data [8]. This latter strategy is used natively by the u-shapelet algorithm.

1.1. Review of u-shapelets

Let us consider a dataset consisting of 4 time series corresponding to birds' calls: 2 corresponding to Olive-sided Flycatcher (green time series)and 2 corresponding to calls of the White-crowned Sparrow (blue time series). When these time series are classified using the Euclidean distance as a measure of dissimilarity (Fig. 1 (a)) the obtained groups are not homogeneous; in other words, we cannot recognize the bird from its calls. However, if we look for characteristic sub-sequences (u-shapelets) to classify the time series, we obtain more homogeneous groups (Fig. 1 (b)).



(a) (b)

Figure 1: Example of clustering of time series using on the one hand the Euclidean distance (a), on the other hand the shapelet (b).

Once this observation has been made, the natural question is how to find sub-sequences that characterize a time series group, that is, sub-sequences that are observed only in a particular time series subgroup. The u-shapelet discovery algorithm answers this question and proceeds as follows: the algorithm takes the length of the pattern as a parameter; on each time series, we drag a window of the same length as the pattern, each new sub-sequence obtained by this process is a candidate pattern.

Among the candidate patterns, we consider as a pattern the subsequence able to divide the time series dataset into two subsets D_A and D_B such that D_A contains all the time series that possess the pattern and D_B all those which do not contain the pattern.

Two other constraints are taken into account in the discovery of patterns: the first one is the ability of the pattern to build subsets that are well separated; the second one is the ability of users to build subsets that are not unbalanced. That is, the size of D_A must be at most k times larger than that of D_B and vice versa.

Definition 1. *(Unsupervised-Shapelet candidate) An unsupervised-shapelet (u-shapelet) candidate S' is any subsequence that has a number of data points less than or equal to the number of data points of the shortest time series in the dataset [10].*

The similarity between a time series and a shapelet is evaluated using a distance function.

Definition 2. *(Sub-sequence distance) The subsequence distance $sdist(S, T)$ between a time series T and a subsequence S is the minimum of the distances between the subsequence S and all possible subsequences of T of length equal to the length of S [10].*

This definition opens the question of which distance measure to use for *sdist*. In general, the ubiquitous Euclidean distance (ED) is used, but it is not appropriate for uncertain time series [6]. In the following section, we introduce a dissimilarity function that is more adapted to uncertainty.

Computing the *sdist* between a u-shapelet candidate and all time series in a dataset creates an orderline:

Definition 3. *(Orderline) An orderline is a vector of subsequence distances $sdist(S', T_i)$ between a u-shapelet candidate S' and all time series T_i in the dataset [10].*

The computation of the orderline is time-consuming. An orderline for a single u-shapelet candidate is computed in time $O(NMlog(M))$ where N is the number of time series in the dataset and M is the average length of the time series. The brute force algorithm for U-shapelets discovery requires K such computations, where K is the number of sub-sequences. The strategy used by [10] in **Scalable Unsupervised Shapelet algorithm** consists in filtering the K candidate segments by considering only those allowing to build balanced groups. This selection is efficiently made thanks to a hash algorithm.

Definition 4. *(Unsupervised-Shapelet) A good u-shapelet candidate S' is a subsequence having the following property: sdist between S' and any time series in one group D_A is significantly smaller than sdist between S' and any time series in another group D_B: $sdist(S', D_A) << sdist(S', D_B)$ [10].*

The assessment of a u-shapelet quality is based on its separation power, which is calculated as follows :

$$gap = \mu_B - \sigma_B - (\mu_A + \sigma_A), \tag{1}$$

where μ_A (resp. μ_B) denotes the mean(sdist(S, D_A)) (resp. the mean(sdist(S, D_B))), and σ_A (resp. σ_B) represents the standard deviation of $sdist(S, D_A)$ (resp. the standard deviation of $sdist(S, D_B)$). If D_A or D_B consists of only one element (or of an insignificant number of elements that cannot represent a separate cluster), the gap score is assigned to zero. This ensures that a high gap scored for a u-shapelet candidate corresponds to a true separation power.

1.2. U-shapelets algorithm for clustering Uncertain Time Series

U-shapelets clustering is a framework introduced in [11], which suggests the clustering of time series using the local properties of their sub-sequences rather than using the global features of the time series [12]. Hence, u-shapelets clustering first computes the set of sub-sequences characteristics of the different categories of time series, then it classifies each time series according to the presence or absence of these typical sub-sequences in it.

Clustering time series with u-shapelets has several advantages. First, u-shapelets clustering is defined for datasets in which time series have different lengths, which is not the case of most techniques described in the literature. Indeed, in many cases, the equal length assumption is implied, and the trimming to equal length is done by exploiting expensive human skills [10]. Secondly, u-shapelets clustering is much more expressive regarding representational power. Indeed, the algorithm works only on time series that can be clustered, namely, that are not outliers.

Furthermore, it is very appropriate to use u-shapelets clustering with uncertain time series because it can ignore irrelevant data and thus, reduce the adverse effects of the presence of uncertainties in the time series. Despite this advantage, it is still highly desirable to take into account the adverse impact of uncertainty during u-shapelet discovery [7].

1.3. Uncertainty and u-shapelets discovery issue

Traditional measures of similarity like the Euclidean distance (ED) or the Dynamic Time Warping (DTW) techniques [13] do not always work well with uncertain time series data. Indeed, they aggregate the uncertainty of each data point of the time series being compared and thus amplify the negative impact of uncertainty. However, ED plays a fundamental role in u-shapelet discovery because it is used to compute the gap (Eq. 1). The discovery of u-shapelet on uncertain time series could thus lead to the selection of a wrong u-shapelet candidate or to assign a time series to the wrong cluster.

In this study, our goal is not to define an uncertain u-shapelet algorithm, but rather to use a dissimilarity function robust to uncertainty to improve the quality of the u-shapelets discovered and thus the clustering quality of uncertain time series.

1.4. Summary of contributions

- We review the state of the art on similarity functions for uncertain time series and evaluate them for the comparison of small, uncertain time series.
- We introduce the Frobenius cOrrelation for uncertain Time series u-Shapelet discovery (FOTS), a new dissimilarity score based on local correlation, which has interesting properties for the comparison of small, uncertain time series, and makes no assumption on the probability distribution of uncertainty in data.
- We *put the source code at the disposal* of the scientific community to allow extension of our work [14].

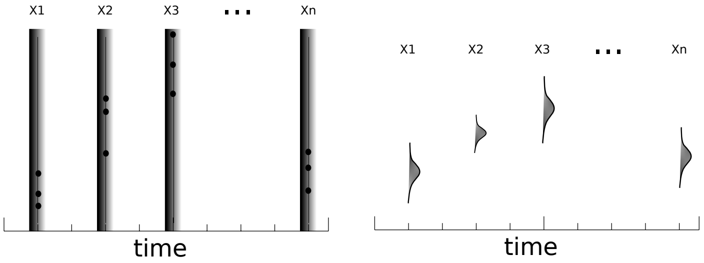
2. Background and Related works

2.1. Background

An Uncertain Time Series (UTS) $X =< X_1, \dots, X_n >$ is a sequence of random variables where X_i is the random variable modeling the unknown real value number at timestamp i. There are two main ways to model uncertain time series: multiset-based model and PDF-based model [7].

. In the **Multiset-based model**, each element $X_i (1 \leq i \leq n)$ of an UTS $X =< X_1, \dots, X_n >$ is represented as a set $\{X_{i,1}, \dots, X_{i,N_i}\}$ of observed values and N_i denotes the number of observed values at timestamp i (Fig. 2a).

. In the **PDF-based model**, each element $X_i, (1 \leq i \leq n)$ of UTS $X =< X_1, \dots, X_n >$ is represented as a random variable $X_i = x_i + X_{e_i}$ (Fig. 2b), where x_i is the exact value that is unknown and X_{e_i} is a random variable representing the error. It is this model that we consider in this work.



(a) Multiset-based model of uncertain time series

(b) PDF-based model of uncertain time series

Figure 2: Uncertain Time Series models

Several similarity measures have been proposed for uncertain time series. They are grouped into two main categories: Traditional similarity measures and uncertain similarity measures.

- Traditional similarity measures such as the Euclidean distance are those conventionally used with time series. They use a single uncertain value at each timestamp as an approximation of the unknown real value [15, 16, 17].
- Uncertain similarity measures use additional statistical information that quantifies the uncertainty associated with each approximation of the real value : this is the case of DUST, PROUD, MUNICH[18]. [8] demonstrated that the performances of uncertain similarity measures associated with the pre-processing of data are higher than those of traditional similarity measurements.

2.2. State of the art on uncertain similarity functions

Uncertain similarity measures can be grouped into two broad categories : deterministic similarity measurements and probabilistic similarity measurements [6].

2.2.1. Deterministic Similarity Measures

Like traditional similarity measures, deterministic similarity measures return a real number as the distance between two uncertain time series. DUST is an example of a deterministic similarity measure.

DUST. [19] Given two uncertain time series $X =< X_1, \ldots, X_n >$ and $Y =< Y_1, \ldots, Y_n >$, the distance between two uncertain values X_i, Y_i is defined as the distance between their true (unknown) values r(X_i), r(Y_i): $dist(X_i, Y_i) = |r(X_i) - r(Y_i)|$. This distance is used to measures the similarity of two uncertain values.

$\varphi(|X_i - Y_i|)$ is the probability that the real values at timestamp i are equal, given the observed values at that instant :

$$\varphi(|X_i - Y_i|) = Pr(dist(0, |X_i - Y_i|) = 0). \tag{2}$$

This similarity function is then used inside the *dust* dissimilarity function:

$$dust(X_i, Y_i) = \sqrt{-log(\varphi(|X_i - Y_i|)) + log(\varphi(0))}. \tag{3}$$

The distance between uncertain time series $X =< X_1, \ldots, X_n >$ and $Y =< Y_1, \ldots, Y_n >$ in $DUST$ is then defined as follows:

$$DUST(X, Y) = \sqrt{\sum_{i=1}^{n} dust(X_i, Y_i)^2}. \tag{4}$$

Chebyshev similarity. Wang et al. [9] showed that a model of uncertain time series inspired by Chebyshev inequality reduced overall computational cost and requires no prior knowledge. Furthermore, they propose a new similarity matching method based on Chebyshev model and analyzed their results by comparing with prior works.

The problem with deterministic uncertain distances like DUST is that their expression varies as a function of the probability distribution of uncertainty, but unfortunately this probability is not always available in time series datasets.

2.2.2. Probabilistic Similarity Measures

Probabilistic similarity measures do not require knowledge of the uncertainty probability distribution. Furthermore, they provide more information about the reliability of the result. There are several probabilistic similarity functions, such as MUNICH, PROUD, PROUDS or Local Correlation.

MUNICH [20] . This distance function is suitable for uncertain time series represented by the multiset based model. The probability that the distance between two uncertain time series X and Y is less than a threshold ε is equal to the number of distances between X and Y, which are less than ε, over the possible number of distances:

$$Pr(distance(X,Y)) \leq \varepsilon = \frac{|\{d \in dists(X,Y)|d \leq \varepsilon\}|}{|dists(X,Y)|} \tag{5}$$

The computation of this distance function is very time-consuming.

PROUD [21]. Let$X =< X_1, ..., X_n >$ and $Y =< Y_1, ..., Y_n >$ be two UTS, each one modeled by a sequence of random variables, the PROUD distance between X and Y is $d(X,Y) = \sum\limits_{i=1}^{n} (X_i - Y_i)^2$. According to the central limit theorem [22], the cumulative distribution of the distances approaches asymptotically a normal distribution:

$$d(X,Y) \propto N(\sum_i E[(X_i - Y_i)^2], \sum_i Var[(X_i - Y_i)^2]) \tag{6}$$

As a consequence of that feature of the PROUD distance, the standard normal distribution table can be used to compute the probability that the normalized distance is lower than a threshold:

$$Pr(d(X,Y)_{norm} \leq \epsilon). \tag{7}$$

A major disadvantage of PROUD is its inadequacy for comparing time series of small lengths like u-shapelets. Indeed, the calculation of the probability that the PROUD distance is less than a value is based on the assumption that it follows **asymptotically** a normal distribution. Thus, this probability will be all the more accurate as the compared time series are long (more than 30 data points).

PROUDS [8]. is an enhanced version of PROUD, which supposes that random variables coming from time series are independent and identically distributed.

Definition 5. *(Normal form of a time series) The normal form of a standard time series $X =< X_1, \dots, X_n >$ is defined as $\hat{X} =< \hat{X}_1, \dots, \hat{X}_n >$ in which for each timestamp i $(1 \leq i \leq n)$, we have:*

$$\hat{X}_i = \frac{X_i - \bar{X}}{S_X},\ \bar{X} = \sum_{i=1}^{n} \frac{X_i}{n},\ S_X = \sqrt{\sum_{i=1}^{n} \frac{(X_i - \bar{X})^2}{(n-1)}}. \tag{8}$$

PROUDS defines the distance between two normalized time series $\hat{X} =< \hat{X}_1 ... \hat{X}_n >$ and $\hat{Y} =< \hat{Y}_1 ... \hat{Y}_n >$ (Definition 5) as follows:

$$Eucl(\hat{X}, \hat{Y}) = 2(n-1) + 2 \sum_{i=1}^{n} \hat{X}_i \hat{Y}_i \tag{9}$$

For the same reasons as PROUD, PROUDS is not suitable for short time series comparison. Another weakness of PROUDS is its assumption that the random variables are independent : this hypothesis is heavy and particularly inappropriate for short time series like u-shapelets. A more realistic hypothesis with time series would be to consider that the random variables constituting the time series are M-dependent. Random variables of a time series are called M-dependent if $X_i, X_{i+1}, ..., X_{i+M}$ are dependent (correlated) and the variables X_i and X_{i+M+1} are independent. However, the M-dependent assumption could make programming PROUDS more complex and its use more difficult because of the choice of the parameter M.

Uncertain Correlation. [7] : Correlation analysis techniques are useful for feature selection in uncertain time series data. Indeed, a correlation indicates the degree of dependency of a feature on other features. Using this information, redundant features can be identified. The same strategy can be useful for u-shapelet discovery. Uncertain correlation is defined as follows :

Definition 6. *(Uncertain time series correlation) Given UTS $X =< X_1, \dots, X_n >$ and $Y =< Y_1, \dots, Y_n >$, their correlation is defined as:*

$$Corr(X, Y) = \sum_{i=1}^{n} \hat{X}_i \hat{Y}_i / (n - 1), \tag{10}$$

where $\hat{X}_i$ and $\hat{Y}_i$ are normal forms of X_i and Y_i (Definition 5), respectively. X_i and Y_i are supposed to be independent continous random variables.

If we know the probability distribution of random variables, it is possible to determine the probability density function associated with the correlation, which will subsequently be used to calculate the probability that the correlation between two time series is greater than a given threshold.

Uncertain correlation has however some limitations :

- It is too sensitive to transient changes, often leading to widely fluctuating scores;
- It cannot capture complex relationship in time series;
- It requires knowledge of the probability distribution function of the uncertainty or to make some assumption on the independence of the random variables contained in time series.

Because of all these limitations, uncertain correlation cannot be used as it is for u-shapelet discovery. The next paragraph presents a generalization of the correlation coefficient that is not an uncertain similarity function but is still interesting for u-shapelet discovery.

Local Correlation. [23] is a generalization of the correlation. It computes a time-evolving correlation score that tracks a local similarity on time series based on a local autocorrelation matrix. The autocorrelation matrix **allows capturing complex relationships** like key oscillatory (e.g., sinusoidal) as well as aperiodic trends (e.g., increasing or decreasing) that are present in times series. The use of autocorrelation matrices, which are computed based on overlapping windows, allows **reducing the sensitivity to transient changes** in time series.

Definition 7. *(Local autocovariance, sliding window). Given a time series X, a sample set of windows with length w, the local autocovariance matrix estimator $\hat{\Gamma}_t$ using a sliding window is defined at time $t \in \mathbb{N}$ as (Eq.11) :*

$$\hat{\mathit{\Gamma}}_t(X, w, m) = \sum_{\tau=t-m+1}^{t} x_{\tau,w} \otimes x_{\tau,w}. \tag{11}$$

where $\boldsymbol{x}_{\tau,\omega}$ is a sub-sequence of the time series of length w and started at τ, $x \otimes y = xy^T$ is the outer product of x and y. The sample set of m windows is centered around time t. We typically fix the number of windows to $m = w$.

Given the estimates $\hat{\Gamma}_t(X)$ and $\hat{\Gamma}_t(Y)$ for the two time series, the next step is to compare them and extract a correlation score. This goal is reached using eigenvectors decomposition; The eigenvectors of the autocorrelation matrices capture the key oscillations and aperiodic trends, even **in short time series**. Thus, the subspaces spanned by the first few (k) eigenvectors are used to locally characterize the behavior of each series. Definition 8 formalizes this notion:

Definition 8. *(LoCo score). Given two series X and Y, their LoCo score is defined by*

$$\ell_t(X, Y) = \frac{1}{2}(\|\boldsymbol{U}_X^T \boldsymbol{u}_Y\| + \|\boldsymbol{U}_Y^T \boldsymbol{u}_X\|) \tag{12}$$

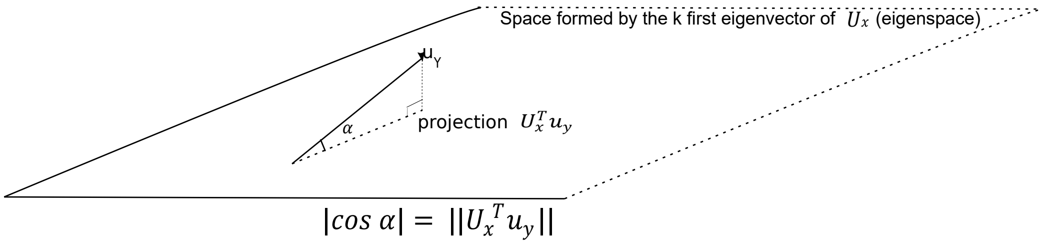

Space formed by the k first eigenvector of U_x (eigenspace)
u_Y
projection $U_x^T u_y$
α
$|cos\ \alpha| = ||{U_x}^T u_y||$

Figure 3: Geometric representation of LoCo similarity.

where $\boldsymbol{U}_X$ and $\boldsymbol{U}_Y$ are the k first eigenvector matrices of the local autocovariance $\hat{\Gamma}_t(X)$ and $\hat{\Gamma}_t(Y)$ respectively, and u_X and u_Y are the corresponding eigenvectors with the largest eigenvalue.

Intuitively, two time series X and Y will be considered as close when the angle α formed by the space carrying the information of the time series X and the vector carrying the information of the time series Y is zero. In other words X and Y will be close when the value of the $cos(\alpha)$ will be 1 (Fig. 3).

The only assumption made for the computation of LoCo similarity is that the mean of time series data points is zero. This could be easily achieved with z-normalization. LoCo similarity function has many interesting properties and does not require to:

- know the probability distribution of the uncertainty,
- assume the independence of the random variables or the length of u-shapelets.

It is therefore interesting for feature selection, but we still need a dissimilarity function to be able to discover u-shapelets. In the next paragraph, we define a dissimilarity function that has the same properties as LoCo and that is robust to the presence of uncertainty.

3. Our Approach

3.1. Dissimilarity function

The LoCo similarity function defined on two time series X and Y approximately corresponds to the absolute value of the cosine of the angle formed by the eigenspaces of X and Y ($|cos(\alpha)|$). A straightforward idea would be to use the $sin(\alpha)$ or α-value as a dissimilarity function but this approach does not work so well; the sinus and the angle are not discriminant enough for eigenvector comparison for clustering purpose. We thus propose the following dissimilarity measure (Definition. 9).

Definition 9. *(FOTS : Frobenius cOrrelation for uncertain Time series u-Shapelet discovery) Given two series X and Y, their FOTS score is defined by*

$$FOTS(X,Y) = \|U_X - U_Y\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{k}(U_X - U_Y)_{ij}^2} \tag{13}$$

where $\|\|_F$ *is the Frobenius norm,* m *is the length of time series,* $\boldsymbol{U}_X$ *and* $\boldsymbol{U}_Y$ *are the k first eigenvector matrices of the local autocovariance* $\hat{\Gamma}_t(X)$ *and* $\hat{\Gamma}_t(Y)$ *respectively.*

Because the FOTS computation is based on the comparison of the k-first eigenvectors of the autocovariance matrices of the time series, it has the same desirable properties of the LoCo similarity function.

3.2. Properties of FOTS score

- It allows to **reduce the sensitivity to transient changes** in time series;
- It is appropriate for the **comparison of short time series**.
- It **allows to capture complex relationships** in time series like the key oscillatory (e.g., sinusoidal) as well as the aperiodic (e.g., increasing or decreasing) trends that are present in times series. The autocovariance matrices capture trends; indeed, positive covariances correspond to similar

variations (growths or decreases) of the two time series : here, either the two values considered are positive or are negative. A negative covariance corresponds to a variation of the two time series in the opposite direction (growth - decrease or decrease - growth): here, one value in the time series is positive and the other is negative. Thus, a sub-matrix of the covariance matrix with positive values corresponds to two sub-sequences of the time series that have a common trend and a submatrix of the autocovariance matrix with alternating positive and negative values identifies oscillations in the time series (Fig. 4). The autocovariance matrix thus makes it possible to capture trends.

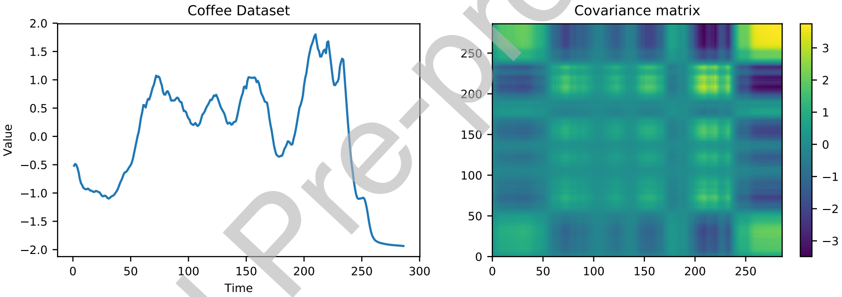


Figure 4: The autocovariance matrix is used to capture trends and oscillations in the time series. For example, the time series of Coffee Dataset decreases between 240 and 300, we can observe a strong correlation and when the time series oscillates between 50 and 250 and we can observe an alternation of light and dark bands in the covariance matrix when $50 \leq x, y \leq 250$.

The auto covariance matrices are symmetric, so there is an eigenvector decomposition of these matrices that captures the main variations of the auto covariance matrix [24, 25] and indirectly the main oscillations and trends in the time series. Indeed, the **Karhunen-Love theorem** [26] stipulates that a time series can be written as a linear combination of the eigenvectors of its covariance matrix, in a manner similar to a Fourrier representation. Thus, these eigenvectors constitute an essential element in the analysis of the structure of time series [27].

Moreover, the FOTS dissimilarity function is **robust to the presence of**

uncertainty due to the eigenvectors decomposition of the autocorrelation matrices of the time series. The robustness of FOTS to the uncertainty is confirmed by the theorem of Hoffman-Wielandt:

Theorem 1. ***(HoffmanWielandt)*** *[28] If X and $X+E$ are $n \times n$ symmetric matrices, then :*

$$\sum_{i=1}^{n} (\lambda_i(X+E) - \lambda_i(X))^2 \leq ||E||_F^2. \tag{14}$$

where $\lambda_i(X)$ is the ith largest eigenvalue of X, and $||E||_F^2$ is the squared of the Frobenius norm of E.

The next section explains how FOTS is integrated in the Scalable Unsupervised Shapelet discovery algorithm.

3.3. Scalable u-shapelets algorithm with the FOTS score

In this section we do not define a new scalable u-shapelets (SUShapelet) algorithm, but we explain how we use the SUShapelet algorithm with the FOTS score (FOTS-SUSh) to deal with uncertainty.

The gap is an essential criterion for the selection of u-shapelet candidates. It is subject to uncertainty because its calculation is based on the Euclidean distance. To remedy this, we propose to use the FOTS score instead of a simple Euclidean distance when calculating the gap in the Scalable u-shapelet algorithm. Algorithm 1 explains how we compute the orderline using the FOTS score; the distance sd_f between the time series s passed as a parameter and all the time series of the data set D is calculated and saved in the variable dis (line 5 and 6). The algorithm returns the normalized distance (line 7). Algorithm 2 calculates the orderline and sorts the time series according to their proximity to the u-shapelet candidate (lines 2 and 3). A u-shapelet is considered present in a time series if its distance to it is less than or equal to a given threshold. The algorithm selects as threshold the ones that produce a cluster with a size between the lower bound lb and the upper bound ub (line 4). The algorithm then searches among the selected thresholds the one that has a maximum gap (line 6 to 11).

Definition 10. *(sub-sequence FOTS dissimilarity) The sub-sequence FOTS dissimilarity* $\boldsymbol{sd_f(S,T)}$ *between a time series* T *and a sub-sequence* S *is the minimum of the FOTS score between the sub-sequence* S *and all possible sub-sequences of* T *of length equal to the length of* S.

Algorithm 1: ComputeOrderline

```
Input: u-shapeletCandidate : s,
time series dataset : D
Output: Distance between the u-shapelet Candidate and all the time
        series of the dataset
function ComputeOrderline($s$, $D$)
    $dis \leftarrow \{\}$
    $s \leftarrow zNorm(s)$
    forall the $i \in \{1, 2, \ldots, |D|\}$ do
        $ts \leftarrow D(i, :)$
        $dis(i) \leftarrow sd_f(s, ts)$
    return $\frac{dis}{|s|}$
```

4. Experimental Evaluation

4.1. Clustering with u-shapelets

There are many ways to cluster time series data described by u-shapelets. In this experiment, the algorithm iteratively splits the data with each discovered u-shapelet: each u-shapelet splits the dataset into two groups D_A and D_B. The time series that belong to D_A are considered as members of the cluster formed by the u-shapelet and are then removed from the dataset. A new u-shapelet search continues with the rest of the data until there are no more time series

Algorithm 2: ComputeGap

```
Input: u-shapeletCandidate : s,
timeseries dataset : D,
lb, ub : lower/upper bound of reasonable number of time series in cluster
Output: gap : gap score
function ComputeGap($s$, $D$, $lb$, $ub$)
    $dis \leftarrow ComputeOrderline(s, D)$
    $dis \leftarrow sort(dis)\ gap \leftarrow 0$
    for $i \leftarrow lb$ to $ub$ do
        $D_A \leftarrow dis \leq dis(i)$, $D_B \leftarrow dis > dis(i)$
        $m_A \leftarrow mean(D_A)$, $m_B \leftarrow mean(D_B)$
        $s_A \leftarrow std(D_A)$, $s_B \leftarrow std(D_B)$
        $currGap \leftarrow m_B - s_B - (m_A + s_A)$
        if $currGap > gap$ then
            $gap \leftarrow currGap$
    return $gap$
```

in the dataset or until the algorithm is no more able to find u-shapelets. As a stopping criterion for the number of u-shapelets extracted, the decrease in the u-shapelet gap score is examined: the algorithm stops when the gap score of the newly-found u-shapelet becomes less than half of the gap score of the first discovered u-shapelet. This approach is a direct implementation of the u-shapelet definition.

Choosing the length N of a u-shapelet. The choice of the length of u-shapelet is directed by the knowledge of the domain to which the time series belongs. As part of these experiments, we tested all numbers between 4 and half the length of the time series. We considered as length of u-shapelet the one allowing to better cluster the time series. Thus, the length used with the Euclidean distance may be different from that used with FOTS. Furtherwork will be done to improve this choice with optimization techniques [29].

Choosing the length w of the windows . The use of overlapping windows for calculating the autocorrelation matrix makes it possible to capture the oscillations present in the time series. During these experiments, we considered that the size of the window is equal to half the length of the u-shapelet.

Choosing the number k of eigenvectors. A practical choice is to fix k to a small value; we use $k = 4$ throughout all experiments. Indeed, key aperiodic trends are captured by one eigenvector, whereas key oscillatory trends manifest themselves in a pair of eigenvectors with similar eigenvalues.

4.2. Evaluation Metric

Different measures for time series clustering quality have been reported, including the Jaccard Score, the Rand Index, the Folkes and the Mallow index, silhouette, correlation, entropy, purity, etc. However, because in our case we have ground truth class labels for the datasets, we can use this external information to evaluate the true clustering quality by using the Rand Index. Moreover, the Rand Index appears to be the most commonly used clustering quality measure [11, 10, 12], and many of the other measures can be seen as minor variants [30]. To appreciate the quality of the u-shapelets found, we use them for a clustering task. The quality of clustering is evaluated from the Rand Index [31], which is calculated as follows:

Let Lc be the cluster labels returned by a clustering algorithm and Lt be the set of ground truth class labels. Let A be the number of time series that are placed in the same cluster in Lc and Lt, B the number of time series in different clusters in Lc and Lt, C the number of time series in the same cluster in Lc but not in Lt and D the number of time series in different clusters in Lc but in same cluster in Lt. The Rand Index is equals to :

$$Rand\,Index = (A + B)/(A + B + C + D) \tag{15}$$

4.3. Comparison with u-shapelet

Similarly to [18], we tested our method on 17 real world datasets, we also extend this test to 46 other datasets, such that 63 datasets taken from the UCR archive [32] are used for our experimental evaluation. The training and testing sets were joined to obtain bigger datasets. FOTS-SUSh performs better than SUSh on 33 datasets, with an average Rand Index of $0.70(+/-0.17)$, SUSh performs better on 27 datasets with an average Rand Index of $0.67(+/-0.15)$, and the two algorithms give the same result on 3 datasets.

Table 2 presents the comparison of the two algorithms (see Appendix).

4.4. Comparison with k-Shape and USLM

k-Shape and USLM are two u-shapelets based clustering algorithms for time series presented in [12]. In this section, we compare the Rand Index obtained by FOTS-SUShapelet and the one obtained by k-Shape and USLM on the only 7 datasets used in [12] and (Table 1). The results of k-Shape and USLM was previously reported in [12]. This comparison shows that in general, FOTS-SUShapelet performs better than k-Shape and USLM on the considering benchmarks.

Table 1: Comparison between k-Shape, USLM and FOTS-SUShapelet

Rand Index	**k-Shape**	**USLM**	**FOTS-SUSh**
CBF	0.74	**1**	0.909
ECG200	0.70	0.76	**0.866**
Fac.F.	0.64	0.79	**0.910**
It. Pow.	0.70	**0.82**	0.50
Lig2	0.65	0.80	**0.911**
Lig.7	0.74	0.79	**0.910**
OSU L.	0.66	0.82	**0.905**

4.5. Discussion

The use of the FOTS score associated with the SUShapelet algorithm allows to discover different u-shapelets from those found by the Euclidean distance. The FOTS-SUSh improves the results of time series clustering because the FOTS score takes into account the intrinsic properties of the time series when searching for u-shapelets and it is robust to the presence of uncertainty. This improvement is particularly significant when the FOTS score is used for the clustering of time series containing several small oscillations. Indeed, these oscillations are not captured by the Euclidean distance but are by the FOTS score whose calculation is based on the autocorrelation matrix. This observation is illustrated by the result obtained on SwedishLeaf dataset (see table 1, appendix).

4.5.1. Time complexity analysis

The Euclidean distance can be computed in time $\mathcal{O}(n)$ and FOTS score is computed in $\mathcal{O}(n^{\omega})$, $2 \leq \omega \leq 3$ due to the time complexity of the eigenvector decompositions [33]. The computation of FOTS score thus is more time consuming than ED (Fig. 5). However, it is competitive to ED for time series of small size, and thus it remains **relevant for u-shapelets research as they are often small**.

4.5.2. Robustness to uncertainty

In order to assess the robustness of FOTS to the presence of uncertainty, we selected two time series from the ItalyPowerDemand dataset and compared them using the Euclidean Distance on one hand and the FOTS score on the other hand (Fig. 6). We then added a white noise that follows a normal distribution of zero mean and 0.1 variance to each of the time series. Then, we recomputed the Euclidean Distance and the FOTS score between the two time series. The absolute value of the difference between the distance obtained with the non-noise time series and that obtained with the noisy time series is called the error. We observe that when the variance associated with white noise increases, the error associated with Euclidean Distance increases, but the error associated with the

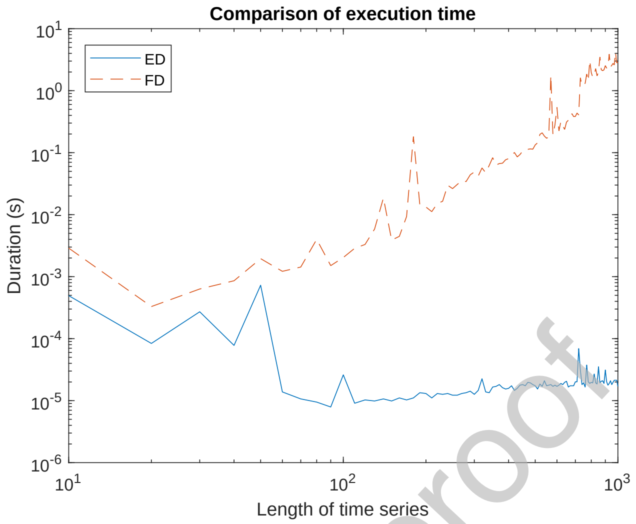


Figure 5: The execution time of ED and FOTS score is a function of the length of time series. The computation time of ED is smaller than that of FOTS.

FOTS score remains almost constant and close to zero (Fig. 6 bottom). This illustrates the robustness of the FOTS score to the presence of uncertainty in the data. Note: In this experiment, noisy time series are considered uncertain time series.

4.5.3. Sensitivity to the choice of the length of the UShapelet

We assume in this work that the SUShapelet clustering algorithm is used by a business expert who knows the length of the pattern to be considered when analyzing time series. However, this is not the case in general. It is therefore important to discuss the choice of this parameter.

The quality of clustering, which is measured in our case by the Rand Index, varies greatly depending on the choice of the UShapelet length; as illustrated in the figure 7 and figure 8. In the case of our experiments, we tested several values for the UShapelets length, ranging from 4 to half the length of the time series and considered a value with a maximum Rand Index. The case where the user does not know a priori the time series classes is not treated here, but it would be possible in a future work to propose a heuristic or a meta-heuristic seeking the UShapelet length that maximizes a metric of clustering quality that is not dependent on the knowledge of time series classes like Silhouette coefficient, single or complete or average linkage [34].

5. General conclusion and Future Work

Our objective during this work was to propose a framework for clustering uncertain time series. To do this, we suggest comparing time series based on sub-sequences called Unsupervised Shapelet (UShapelet) and we propose a dissimilarity function robust to uncertainty, helping to discover UShapelet on uncertain time series. The calculation of this dissimilarity function called FOTS is based on the comparison using the Frobenius distance of the eigenvectors of the autocorrelation matrices of the time series and it has interesting properties: it is not very sensitive to transient changes, it allows capturing complex relationships between time series such as oscillations and trends and it is particularly suitable for comparing short time series like UShapelet due to its high time complexity $\mathcal{O}(n^{\omega})$, $2 \leq \omega \leq 3$. The framework we present was evaluated on a clustering task of 63 data sets from the literature and showed an improvement in the clustering quality measured from Rand Index. However, we noted that the quality of the results obtained is sensitive to the variation of the length of the sub-sequence considered. As a perspective to this work, we are proposing to conceive a heuristic or a metaheuristic, allowing to choose wisely the length of the sub-sequences allowing to group the time series as well as possible, we are also proposing to reduce the computation time of the FOTS dissimilarity function based on sequential learning principles [35]. It would also be interesting to evaluate the performance of this model on a time series classification task.

Acknowledgment

We thank the French Ministry of Higher Education, Research, and Innovation, for funding this work. This work was also partly supported by the Labex project IMobS3. We also thank Liudmila Ulanova, Nurjahan Begum, Eamonn Keogh and Anthony Bagnall, Jason Lines, William Vickers, Eamonn Keogh, the UEA & UCR Time Series Classification Repository, www.timeseriesclassification.com for the considerable work they have done, which provides us with several datasets and source codes. We also thank Nsaibirni Robert and Givry Jean Jacques for reviewing and correcting the paper. We finally thank the reviewers who allowed us to improve the quality of this work through their comments and suggestions.

References

[1] G. B. Folland, A. Sitaram, The uncertainty principle: a mathematical survey, Journal of Fourier analysis and applications 3 (3) (1997) 207–238.

[2] S. Thayer, M. Trivedi, Residual uncertainty in three-dimensional reconstruction using two-planes calibration and stereo methods, Pattern Recognition 28 (7) (1995) 1073–1082.

[3] N. B. Rizvandi, J. Taheri, R. Moraveji, A. Y. Zomaya, A study on using uncertain time series matching algorithms for MapReduce applications, Concurrency and Computation: Practice and Experience 25 (12) (2013) 1699–1718. doi:10.1002/cpe.2895.

[4] J. Hwang, Y. Kozawa, T. Amagasa, H. Kitagawa, GPU Acceleration of Similarity Search for Uncertain Time Series, in: 2014 17th International Conference on Network-Based Information Systems, IEEE, 2014, pp. 627–632. doi:10.1109/NBiS.2014.89.

[5] K. Rehfeld, J. Kurths, Similarity estimators for irregular and age-uncertain time series, Climate of the Past 10 (1) (2014) 107–122. doi:10.5194/cp-10-107-2014.

[6] M. Orang, N. Shiri, An experimental evaluation of similarity measures for uncertain time series, in: Proceedings of the 18th International Database Engineering & Applications Symposium on - IDEAS '14, ACM Press,

New York, New York, USA, 2014, pp. 261–264. doi:10.1145/2628194.2628207.

[7] M. Orang, N. Shiri, Correlation analysis techniques for uncertain time series, Knowledge and Information Systems 50 (1) (2017) 79–116. doi:10.1007/s10115-016-0939-7.

[8] M. Orang, N. Shiri, Improving performance of similarity measures for uncertain time series using preprocessing techniques, in: Proceedings of the 27th International Conference on Scientific and Statistical Database Management - SSDBM '15, ACM Press, New York, New York, USA, 2015, pp. 1–12. doi:10.1145/2791347.2791385.

[9] W. Wang, G. Liu, D. Liu, Chebyshev Similarity Match between Uncertain Time Series, Mathematical Problems in Engineering 2015 (2015) 1–13. doi:10.1155/2015/105128.

[10] L. Ulanova, N. Begum, E. Keogh, Scalable clustering of time series with u-shapelets, in: Proceedings of the 2015 SIAM International Conference on Data Mining, SIAM, 2015, pp. 900–908.

[11] J. Zakaria, A. Mueen, E. Keogh, Clustering time series using unsupervised-shapelets, in: 2012 IEEE 12th International Conference on Data Mining (ICDM), IEEE, 2012, pp. 785–794.

[12] Q. Zhang, J. Wu, H. Yang, Y. Tian, C. Zhang, Unsupervised Feature Learning from Time Series., in: IJCAI, 2016, pp. 2322–2328.

[13] D. Folgado, M. Barandas, R. Matias, R. Martins, M. Carvalho, H. Gamboa, Time Alignment Measurement for Time Series, Pattern Recognition 81 (2018) 268–279.

[14] V. S. Siyou Fotso, E. Mephu Nguifo, P. Vaslin, fots-sush experimentations (2018).
URL https://sites.google.com/view/fots-sush/accueil

[15] C. Y. Zhou, Y. Q. Chen, Improving nearest neighbor classification with cam weighted distance, Pattern Recognition 39 (4) (2006) 635–645.

[16] J. Zhao, L. Itti, shapeDTW: Shape Dynamic Time Warping, Pattern Recognition 74 (2018) 171–184.

[17] K. Ø. Mikalsen, F. M. Bianchi, C. Soguero-Ruiz, R. Jenssen, Time series cluster kernel for learning similarities between multivariate time series with missing data, Pattern Recognition 76 (2018) 569–581.

[18] M. Dallachiesa, B. Nushi, K. Mirylenka, T. Palpanas, Uncertain time-series similarity: return to the basics, Proceedings of the VLDB Endowment 5 (11) (2012) 1662–1673.

[19] K. Murthy, S. R. Sarangi, Generalized notion of similarities between uncertain time series, uS Patent 8,407,221 (Mar. 26 2013).

[20] J. Aßfalg, H.-P. Kriegel, P. Kröger, M. Renz, Probabilistic Similarity Search for Uncertain Time Series, in: SSDBM, Springer, 2009, pp. 435–443.

[21] M.-Y. Yeh, K.-L. Wu, P. S. Yu, M.-S. Chen, PROUD: a probabilistic approach to processing similarity queries over uncertain data streams, in: Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology, ACM, 2009, pp. 684–695.

[22] J. Hoffmann-Jørgensen, G. Pisier, The law of large numbers and the central limit theorem in banach spaces, The Annals of Probability (1976) 587–599.

[23] S. Papadimitriou, J. Sun, S. Y. Philip, Local correlation tracking in time series, in: Data Mining, 2006. ICDM'06. Sixth International Conference on, IEEE, 2006, pp. 456–465.

[24] M. Ghil, M. Allen, M. Dettinger, K. Ide, D. Kondrashov, M. Mann, A. W. Robertson, A. Saunders, Y. Tian, F. Varadi, et al., Advanced spectral methods for climatic time series, Reviews of geophysics 40 (1) (2002) 3–1.

[25] S. Papadimitriou, F. Li, G. Kollios, P. S. Yu, Time series compressibility and privacy, in: Proceedings of the 33rd international conference on Very large data bases, VLDB Endowment, 2007, pp. 459–470.

[26] S. Pranesh, D. Ghosh, Faster computation of the karhunen–loève expansion using its domain independence property, Computer Methods in Applied Mechanics and Engineering 285 (2015) 125–145.

[27] N. Golyandina, V. Nekrutkin, A. A. Zhigljavsky, Analysis of time series structure: SSA and related techniques, Chapman and Hall/CRC, 2001. doi:10.1201/9780367801687.

[28] R. Bhatia, T. Bhattacharyya, A generalization of the Hoffman-Wielandt theorem, Linear Algebra and its Applications 179 (1993) 11–17. doi:10.1016/0024-3795(93)90318-I.

[29] V. S. Siyou Fotso, E. Mephu Nguifo, P. Vaslin, Grasp heuristic for time series compression with piecewise aggregate approximation, RAIRO-Operations Research 53 (1) (2019) 243–259.

[30] M. Halkidi, Y. Batistakis, M. Vazirgiannis, On clustering validation techniques, Journal of intelligent information systems 17 (2-3) (2001) 107–145.

[31] W. M. Rand, Objective criteria for the evaluation of clustering methods, Journal of the American Statistical association 66 (336) (1971) 846–850.

[32] H. A. Dau, E. Keogh, K. Kamgar, C.-C. M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, Yanping, B. Hu, N. Begum, A. Bagnall, A. Mueen, G. Batista, Hexagon-ML, The UCR Time Series Classification Archive (July 2019).
URL www.cs.ucr.edu/~eamonn/time_series_data/

[33] V. Y. Pan, Z. Q. Chen, The complexity of the matrix eigenproblem, in: Proceedings of the thirty-first annual ACM symposium on Theory of computing, ACM, 1999, pp. 507–516.

[34] E. Achtert, S. Goldhofer, H.-P. Kriegel, E. Schubert, A. Zimek, Evaluation of Clusterings – Metrics and Visual Support, in: 2012 IEEE 28th International Conference on Data Engineering, 2012, pp. 1285–1288.

[35] D. Calandriello, Efficient Sequential Learning in Structured and Con- 635 strained Environments, Ph.D. thesis, Inria Lille Nord Europe-Laboratoire CRIStAL-Université de Lille (2017).

Appendix

Table 2: Comparaison of the Rand Index of FOTS-SUSh and SUSh on 63 datasets. SLength stand for UShapelet Length

Dataset	**SLength**	**RI_FOTS**	**SLength**	**RI_Sh**
50Words	5	**0.88**	6	0.81
Adiac	6	**0.91**	4	0.79
Beef	28	**0.91**	7	0.89
Car	19	**0.72**	6	0.71
CBF	6	**0.91**	39	0.57
Coffee	6	**0.90**	24	0.78
ECG200	5	**0.87**	24	0.71
FaceAll	6	**0.91**	4	0.90
FaceFour	7	**0.91**	7	0.85
FISH	6	**0.90**	6	0.77
GunPoint	6	**0.89**	38	0.71
Ligthning2	6	**0.91**	5	0.79
Ligthning7	6	**0.91**	37	0.71
OliveOil	6	**0.91**	24	0.71
OSULeaf	6	**0.91**	5	0.84
SwedishLeaf	6	**0.91**	5	0.5

Continued on next page

Table 2 – *Continued from previous page*

Dataset	**SLength**	**RI_FOTS**	**SLength**	**RI_Sh**
Synthetic_control	5	**0.90**	25	0.72
ArrowHead	6	**0.64**	50	0.61
BeetleFly	43	0.57	18	**0.61**
BirdChicken	23	0.55	12	**0.61**
Computers	11	0.51	13	**0.52**
Cricket_X	6	**0.84**	9	0.72
Cricket_Y	6	**0.84**	8	0.79
Cricket_Z	6	**0.84**	10	0.72
DiatomSizeReduction	15	0.69	29	**0.80**
DistalPhalanxOutlineAgeGroup	12	0.61	6	**0.73**
DistalPhalanxOutlineCorrect	15	0.52	5	**0.53**
DistalPhalanxTW	10	0.81	5	**0.86**
Earthquakes	7	0.47	20	**0.59**
ECGFiveDays	14	0.51	32	**0.86**
FacesUCR	6	**0.84**	10	0.33
Ham	19	0.51	43	0.51
Herring	22	0.52	9	0.52
ItalyPowerDemand	8	0.50	7	**0.52**
LargeKitchenAppliances	5	0.59	10	**0.62**
Meat	36	0.65	35	**0.83**
MedicalImages	7	0.66	6	**0.67**
MiddlePhalanxOutlineAgeGroup	15	**0.74**	4	0.71
MiddlePhalanxOutlineCorrect	9	**0.51**	37	0.49
MiddlePhalanxTW	15	0.76	5	**0.82**
MoteStrain	12	0.51	24	**0.53**
PhalangesOutlinesCorrect	9	**0.52**	15	0.50
Plane	6	0.79	13	**0.94**

Continued on next page

Table 2 – *Continued from previous page*

Dataset	**SLength**	**RI_FOTS**	**SLength**	**RI_Sh**
ProximalPhalanxOutlineAgeGroup	13	0.74	5	**0.80**
ProximalPhalanxOutlineAgeCorrect	11	0.57	40	**0.59**
ProximalPhalanxTW	13	0.81	5	**0.85**
RefrigirationDevices	10	**0.63**	10	0.61
ScreenType	21	0.61	10	**0.62**
ShapesAll	6	**0.90**	11	0.77
ShapeletSim	10	1.00	6	1,00
SmallKitchenAppliances	4	**0.64**	32	0.58
SonnyAIBORobotSurface	9	0.53	27	**0.84**
Strawberry	10	0.51	6	**0.56**
ToSegmentation	40	0.50	32	**0.65**
Trace	10	0.83	15	**1,00**
ElectricDevices	6	**0.36**	5	0.3
SonnyAIBORobotSurfaceII	15	0.53	15	**0.64**
ToeSegmentation2	6	0.48	15	**0.69**
TwoLeadECG	15	0.55	10	**0.85**
Wine	10	0.52	6	**0.56**
WordSynonyms	6	**0.84**	6	0.48
Worms	10	**0.57**	15	0.42
WormsTwoClass	6	0.50	6	**0.51**

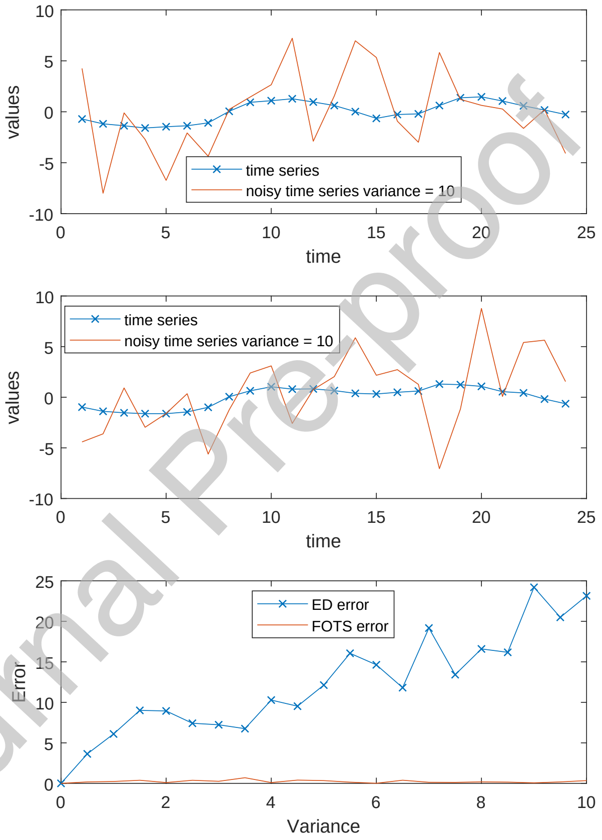
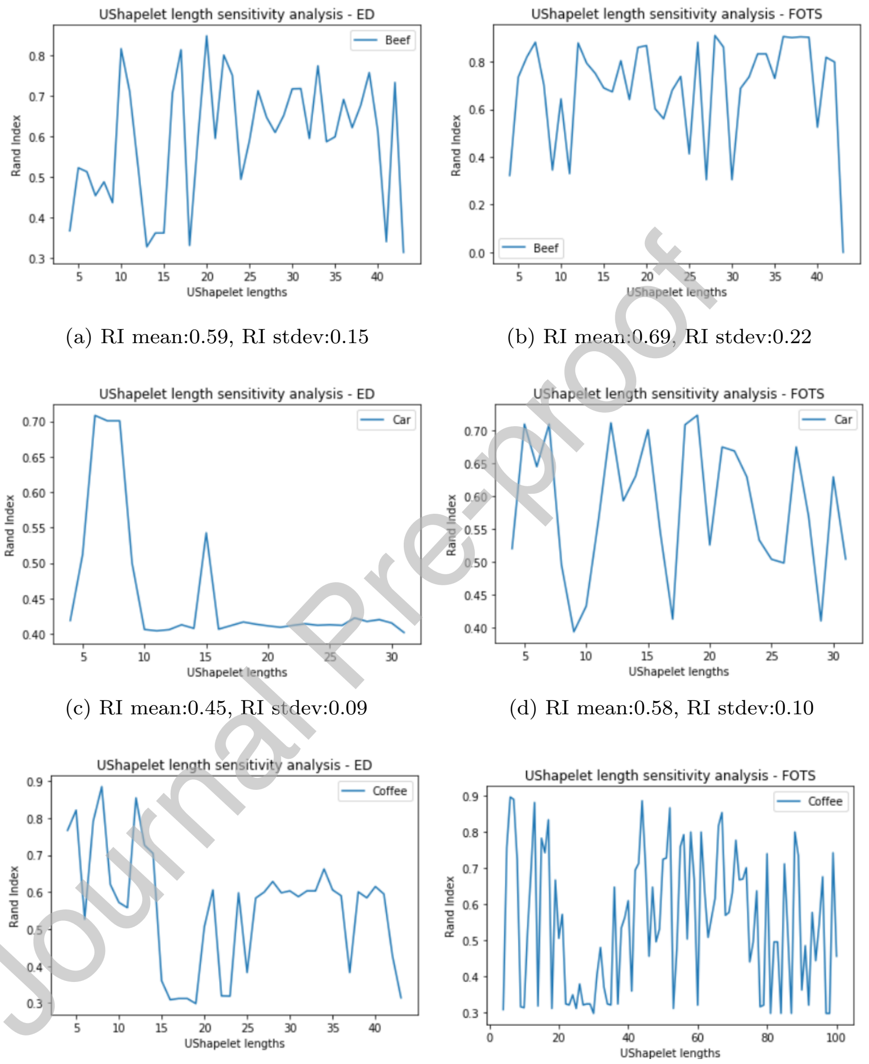


Figure 6: Sensitivity of Euclidean Distance and FOTS to the presence of uncertainty. The two time series come from ItalyPowerDemand dataset



(a) RI mean:0.59, RI stdev:0.15

(b) RI mean:0.69, RI stdev:0.22

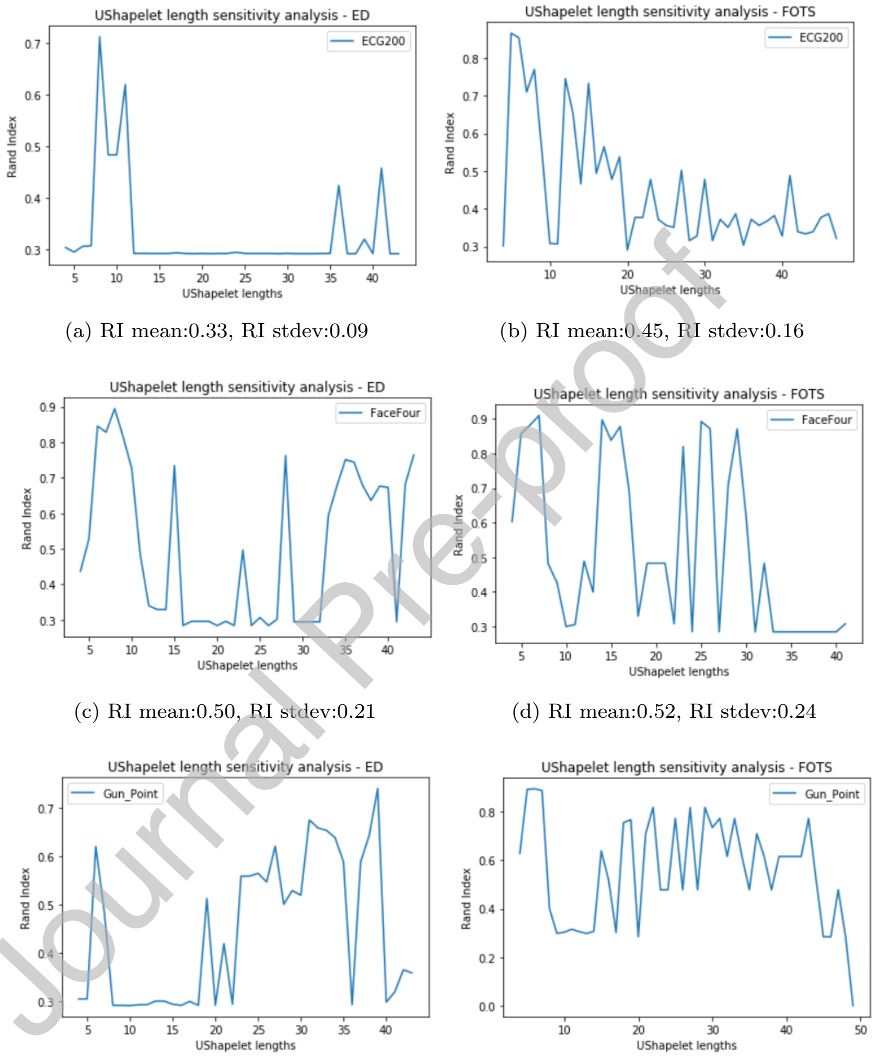
(c) RI mean:0.45, RI stdev:0.09

(d) RI mean:0.58, RI stdev:0.10

(e) RI mean:0.56, RI stdev:0.16

(f) RI mean:0.55, RI stdev:0.19

Figure 7: Analysis of the sensitivity of the SUShapelet algorithm to the choice of the length of the UShapelet, case of the Beef, Car, Coffee data set. RI stands for Rand Index, stdev stands for Standard Deviation.



(a) RI mean:0.33, RI stdev:0.09

(b) RI mean:0.45, RI stdev:0.16

(c) RI mean:0.50, RI stdev:0.21

(d) RI mean:0.52, RI stdev:0.24

(e) RI mean:0.44, RI stdev:0.15

(f) RI mean:0.55, RI stdev:0.21

Figure 8: Analysis of the sensitivity of the SUShapelet algorithm to the choice of the length of the UShapelet, case of the ECG200, FaceFour, Gun_Point data set. RI stands for Rand Index, stdev stands for Standard Deviation.

***Auth**

Vanel Steve Siyou Fotso received his Ph.D. degree from the University Clermont Auvergne (UCA), France. He is currently a researcher in Datamining field and served as the team leader of the R&D division of Master Data Solution a start-up located at Levallois-Perret, France. His major research interests include knowledge discovery in time series and uncertainty management.

Engelbert Mephu Nguifo is a full professor of computer science at University Clermont Auvergne (UCA), France. At UCA, he has served as Vice-President of the mathematical and computer science department (2012-2016). He is leading research on complex data mining and machine learning in the joined University-CNRS laboratory LIMOS (Laboratory of Computer Science, Modelisation and Optimization), where he is co-chair of the Information and Communication Systems research group. His research interests include formal concept analysis, artificial intelligence, machine learning, complex data mining, pattern recognition, bioinformatics, big data, and knowledge representation. Mephu Nguifo has a Ph.D. in computer science from the University of Montpellier. He has published more than hundred technical papers in majors journals and conferences, and was advisor of more than ten PhD students currently in academic position. He was principal investigator of several international research project grants. He is member of the steering committee of international conference on Concept Lattices and their Applications (CLA), and has served as PC Chair of CLA in 2006, as well as French conference on Machine Learning (CAp) in 2010. He has co-organized several workshops of majors conferences (ECAI, IJCAI, ECML/PKDD, VLDB) on several research topics (Bioinformatics and AI, Evolving Graphs, Concept Lattices). He is an ACM **Senior** member, and member of French AI, bioinformatics, data mining, and classification associations (AFIA, SFBI, EGC). He is member of the executive board of the French Association on Artificial Intelligence (AFIA). He is also member of the executive board of the French CNRS research group on Artificial Intelligence (GDR IA).

Dr Philippe VASLIN (M) received his Ph.D. in Sport Sciences (speciality: Biomechanics) at Bordeaux University (France) in 1993 and he joined Blaise Pascal University (UBP, France) as an Assistant Professor in 1994. Before completing his thesis, he worked as a Research Engineer in two R&D offices (INRES, Bordeaux; CRIPS, Belfort) where he was in charge of mechanical and field testing, comparative studies, and research reporting on various products. Since 1998, he has been mainly working on the biomechanics of manual wheelchair locomotion at UCA-LIMOS. This research has been financially supported by two French national projects (ANAES 1999, ANR TecSan 2006). Beside this main topic, he has also been involved in research projects on sport biomechanics with Poitiers University (France) and on the attitude control of high-speed robotic wheeled vehicles during a ballistic phase with SIGMA (UCA, France). He is the author or co-author of about one hundred of national and international scientific articles, book chapters and communications.

Declaration of Interest Statement

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: